

A METHOD AND APPARATUS FOR CONTROLLING DISPLACEMENTS OF A
MOVING PORTION OF A MULTI-AXIS ROBOT

The invention relates to a method and to apparatus
for controlling displacements of a moving portion of a
5 multi-axis robot along a path.

In the field of controlling multi-axis robots, it is
known to control a multi-axis robot by means of PID
(proportional-integer-derivative) type servo-control
loops acting on position, speed, and current, in order to
10 cause the terminal portion of a robot, which may be
carrying a tool or a grip, to move along a path of
determined shape and velocity profile. It is also known
to control such a robot by reducing the stiffness of the
servo-control on each axis in order to enable the
15 position of the arm to be varied as a function of the
forces of interaction between the moving portions of the
robot and their environment. In particular, it is known,
e.g. from US-A-5 742 138, to use parameters to set the
stiffness of a multi-axis robot arm in a rectangular
20 coordinate system, to measure a position error, to
calculate a force, and to correct the position of the
moving portion of the robot so as to obtain a certain
amount of flexibility in its terminal member. Such
flexibility can be used for performing operations of
25 handling parts for purposes such as polishing, deburring,
or extracting from a press, in which a contact force can
oppose the movement programmed for the robot.

It is also known to give a setpoint force in a
rectangular coordinate system in order to implement an
30 operation at a controlled force level along a path, e.g.
for deburring or for polishing.

In known systems, provision is sometimes made for
the path-programming interface to be capable of
specifying parameters such as the velocity or the
35 acceleration of the moving portion along its path, said
parameters determining the position of the terminal

portion of the robot arm along its path as a function of time.

It is also known from US-A-4 874 997 to control a brushless motor for a robot joint numerically by using pulse-width modulation.

In those known systems, it is not easy to program a multi-axis robot so that the displacement path of its terminal portion can be controlled both in velocity and acceleration on the one hand, and so that a force tangential to the path at the point of contact between the moving portion and its environment is also controlled on the other hand, while simultaneously complying with velocity and acceleration values that are compatible with the mechanical structure of the robot.

EP-A-0 349 291 also discloses diverting a robot from the path it has taken in order to comply with a force setpoint as a function of the real shape of a part to be processed. In that example, force control is privileged over following the path, and that is not compatible with certain utilizations in which it is essential to follow the path accurately.

The invention seeks more particularly to satisfy those drawbacks by proposing a method in which the interactions between the environment and the moving portion of a robot, in particular a part or a tool carried thereby, are controlled accurately.

In this spirit, the invention relates to a method of controlling displacements of a moving portion of a multi-axis robot along a path, in which there are provided steps consisting in:

- providing movement instructions to a path generator, the instructions including at least information relating to the shape of the path and to force setpoints;
- calculating an external force signal representing at least one component of the force exerted by the moving portion on its environment;

- acting at a predetermined sampling frequency to provide the external force signal to the path generator;

- calculating, with the path generator and at a predetermined sampling frequency, movement setpoints

5 along the path in such a manner as to minimize the difference between the projection of the external force onto the tangent of the path and the projection of the setpoint onto the tangent; and

- delivering the movement setpoints to a servo-

10 control means enabling at least one axis of the robot to be set into movement in compliance with the movement setpoints.

The tangential force servo-control thus makes it possible to conserve the shape of the path while adapting
15 the velocity of the moving portion to said path.

According to advantageous aspects, a method of controlling displacements of a moving portion of a multi-axis robot may incorporate one or more of the following characteristics:

20 • The external force signal is calculated from information representing the current flowing in at least one actuator of the robot.

- A step is provided consisting in using a dynamic model of said robot while calculating said external force
25 signal.

- A step is provided consisting in supplying said path generator with at least one velocity limit value and/or at least one acceleration limit value for taking into account while calculating said movement setpoints,
30 such that said setpoints comply with said limit value(s).

The invention also provides an apparatus enabling the above-described method to be implemented, and more specifically, an apparatus comprising:

- a path generator suitable for calculating movement
35 setpoints as a function of movement instructions including at least information relating to the shape of the path and to its force setpoints; and

· a force estimator suitable for generating an external force signal representing at least one component of the force exerted by said moving portion on its environment and for delivering said signal to said path generator at a predetermined sampling frequency, where said path generator is suitable for calculating said movement setpoints along said path at a predetermined sampling frequency in such a manner as to minimize the difference between the projection of the external force on the tangent to the path and the projection of the force setpoint onto said tangent, said movement setpoints being delivered to a servo-control means enabling at least one axis of said robot to be set into movement.

Advantageously, the apparatus comprises program interpreter means suitable for executing programs containing movement instructions enabling at least the shape of the path and force setpoints to be specified.

The invention can be better understood and other advantages thereof appear more clearly in the light of the following description of an implementation of a method in accordance with the principle of the invention, given purely by way of example and described with reference to the accompanying drawings, in which:

· Figure 1 is a diagram showing the principle of a multi-axis robot moving a milling tool along a path while implementing a method in accordance with the invention;

· Figure 2 is a diagram showing the principle whereby forces are resolved at a moving portion of the Figure 1 robot;

· Figure 3 is a block diagram showing the general operation of the invention;

· Figure 4 is a block diagram showing the transfer of position setpoints to the corresponding servo-control system for each motor;

· Figure 5 is a block diagram showing the servo-control for each motor;

• Figure 6 shows the method used for estimating external forces; and

• Figure 7 shows the algorithm used by the path generator for generating movement setpoints.

5 The robot R shown in Figure 1 is a multi-axis robot having six joints. It can move with six degrees of freedom. The robot R can be controlled in Cartesian mode, in which case its degrees of freedom can be three
10 degrees of freedom in translation along the directions of three axes X, Y, and Z, and three degrees of freedom in rotation R_x , R_y , and R_z about said axes. The robot may also be controlled in joint mode, in which case the
15 degrees of freedom can be six rotations R_1 , R_2 , R_3 , R_4 , R_5 , and R_6 about the six hinge axes X_1 , X_2 , X_3 , X_4 , X_5 , and X_6 of the joints.

The robot is associated with a control unit U which controls this operation during stages of training and utilization.

20 A manual control console B may also be used during training stages.

In the example shown, the robot R carries a tool O such as a milling cutter, which is to be displaced along a path T.

25 As can be seen more particularly in Figure 2, it is considered that the tool O is displaced by the robot R along the edge B_1 of a part to be milled. The path T is generally parallel to said edge. The force exerted by the tool O against the edge B_1 is written F. The
30 component of this force that is tangential to the path T is written F_T , and the component of said force that is normal to the path T is written F_N .

In accordance with the invention, the value of the tangential component F_T is used in the method of controlling the displacements of the tool O.

35 In the description below, F is used broadly and represents the vector of the torques in the joints of the robot arm or the torsor of external forces including the

Cartesian forces F_x , F_y , F_z , and the Cartesian torques M_x , M_y , M_z .

As shown in Figure 3, the programming environment 100 enables the user of the system to program the robot in such a manner that it performs the desired operations. These operations are described by programs. The programming environment enables the successive positions to be reached by the robot to be trained, it enables programs to be written for describing the sequencing of movements, it enables the programs to be debugged, it enables them to be launched, to be stopped, etc.

In order to execute the programs, the programming environment makes use of a program interpreter 200. This interpreter executes the programs for implementing a task by the robot. Amongst other things, the programs executed contain instructions 300 known as movement instructions. They describe the types of movement to be performed by the robot (straight line, circle, ..., etc.), and also the various parameters characterizing them (target point, velocity, force, ...).

The movement instructions 300 are transferred to the path generator 400. The path generator serves to execute them, i.e. it serves to generate movement setpoints 500 regularly (typically once every 4 milliseconds (ms)), as a function of movement type and movement parameters. These movement setpoints 500 represent the path that is to be followed by the robot at each instant. The servo-controlled industrial robot 600 comprises an industrial arm having six axes together with its servo-control system. The servo-control system is designed in such a manner as to control the six motors of the robot (one for each axis) in such a manner that the path actually followed by the robot follows the path defined by the movement setpoint 500 as closely as possible.

In certain applications, the terminal member of the industrial robot 600 interacts mechanically with its environment: for example, the part it is milling. The

external force estimator 700 constitutes means for obtaining a signal referred to as the external force 800 that is representative of the value and the direction of the forces exerted by the robot on its environment. This
 5 signal is delivered to the path generator 400 in real time (typically once every 4 milliseconds (ms)). The path generator uses this return information to modify the movement setpoints so as to regulate the interaction forces.

10 The programming environment includes a control console that enables the robot to be displaced in an interactive mode, so as to make it possible to train the various positions that are to be taken by the robot. It also serves as a programming interface for writing and
 15 debugging programs. The language used for writing such programs is a text language, but other types of language (e.g. graphics languages) could be used in the context of the invention.

The language used contains instructions of the kind
 20 that are common in robot programming languages, serving to specify different types of movement, to control their velocities and accelerations, and also to control how they are sequenced.

By way of example, there follows a description of
 25 two other instructions that relate to the invention.

The instruction:

MOVEJF (position, tool, params, force)

specifies that the robot must displace the tool it is carrying and as defined by the parameter "tool", from the
 30 final position of the preceding movement to the position defined by the "position" parameter. The displacement is performed in joint mode, i.e. there is a constant proportionality ratio between the displacements of each axis during the movement. The "params" parameter
 35 specifies the velocities and the accelerations that are not to be exceeded during the movement. It also specifies whether the movement is to be run on into the

following movement. The "force" parameter specifies the value desired for the force that is tangential to the path during the movement. Insofar as the limits on velocity and acceleration as defined by "params" are not
 5 reached, the robot should adapt its displacement along the path in such a manner as to maintain the tangential force at a value that is as close as possible to the value specified by the "force" parameter.

The instruction:

10 MOVELF (position, tool, params, force)
 is similar. The only difference is that the movement needs to be performed in a straight line.

Similar instructions can easily be imagined for defining paths that are circular, or that are defined by
 15 splines, or otherwise,

It is important for the invention that these instructions define firstly the shape of the path that the robot is to follow, and secondly the value of the tangential force that the robot is to exert during its
 20 displacement.

The program interpreter 200 serves to execute the programs generated using the development environment. To execute the movement instructions, it verifies the validity of the parameters specified by the user, and
 25 then transmits the movement instruction 300 to the path generator 400. The movement instructions 300 contain information coming from values specified in the program, namely:

• 310 - Definition of the shape of the path: a flag
 30 indicates the type of path (joint or straight line). If the type is "joint", then the joint coordinates q_0 of the starting point and q_1 of the target point are given. If the type is "straight line", then the Cartesian coordinates x_0 of the starting point and x_1 of the target
 35 point are given;

- 320 - Tangential force setpoint: this is the desired value des for the norm of the Cartesian force exerted by the end of the robot on its environment;

- 330 - Velocity limit: the limiting value v_{max} for the norm of the Cartesian velocity at the end of the robot; and

- 340 - Acceleration limit: the limiting value a_{max} for the norm of the Cartesian acceleration at the end of the robot.

10 In the embodiment described herein, the movement commands 500 are constituted by the angular positions q_{des} of the six motors. They are updated by the path generator once every 4 ms. Each of the six position setpoints q_{des1} , q_{des2} , ..., q_{des6} , is transferred to the
15 corresponding servo-control system for each of the motors 601, 602, ..., 606, as shown in Figure 4.

Each motor is fitted with a position and velocity sensor. The servo-control of each motor is of the proportional-integer-derivative (PID) type. In Figure 5,
20 $q_{des,j}$ is the position setpoint on the axis j , $q_{mes,j}$ is the measured position, $K_{p,j}$, $K_{i,j}$, and $K_{d,j}$ are the respective gains of the PID, $I_{des,j}$ is the current setpoint, $I_{mes,j}$ is the measured current, and p is the Laplace variable.

25 Other position servo-control techniques can also be used in the context of the present invention. For example, servo-control may be performed by the decoupling and linearization technique. Under such circumstances, the servo-control is no longer performed axis by axis.
30 In the present example, the movement setpoints 500 are the angular positions desired for each motor. It is also possible to use the position and the velocity desired for the terminal member, as expressed in Cartesian coordinates. The movement setpoints may also be
35 successive position increments, or the velocities desired for each motor at each instant. The point that is important for the invention is that the path generator

400 acts in real time to calculate the path to be followed by the robot and that a servo-control system ensures that the robot follows that path as closely as possible.

5 Numerous means are available for estimating the force exerted by the robot on the outside in the context of the present invention. The most simple consists in placing a force sensor at the end of the robot. Under such circumstances, the outside force 800 is expressed in
10 Cartesian coordinates. It is also possible to use force sensors or torque sensors placed at various locations on the robot.

 In the implementation proposed and shown in Figure 6, a solution is desired that enables the external
15 force to be estimated by using position servo-control variables.

 The method used requires a value 701 representative of the current I flowing in each motor. In this example, this is equal to the measured current I_{mes} used in the
20 servo-control system.

$I = I_{mes} = [I_{mes,1} \ I_{mes,2} \ \dots \ I_{mes,6}]$

 It would be equally possible to use the current control signal, i.e.:

$I = I_{des} = [I_{des,1} \ I_{des,2} \ \dots \ I_{des,6}]$

25 or any other measured or calculated variable representative of the currents flowing in the motors.

 The motors used are brushless motors. The simplest model 702 for a motor of this type consists in considering that the delivered torque C_m 703 is
30 proportional to the current flowing through the motor, i.e.:

$$C_m = K_t \cdot I$$

 In this expression, I and C_m are vectors containing information relating to each of the axes, and K_t is a
35 diagonal matrix. More complex models could be used as 702.

Furthermore, the method used requires signals that are representative of the movements of the robot 711, e.g. the positions, velocities, and accelerations of each of the motors as represented respectively by the vectors \underline{q} , $d\mathbf{q}/dt$, and $d^2\mathbf{q}/dt^2$. In this example, these values are deduced from the measurements of the sensors used by the servo-control system and from their derivatives.

$$\begin{aligned} \mathbf{q} &= \mathbf{q}_{mes} \\ d\mathbf{q}/dt &= d(\mathbf{q}_{mes})/dt \\ 10 \quad d^2\mathbf{q}/dt^2 &= d^2(\mathbf{q}_{mes})/dt^2 \end{aligned}$$

It would also be possible to use the position setpoints and their derivatives: \mathbf{q}_{de} , $d(\mathbf{q}_{des})/dt$, $d^2(\mathbf{q}_{mes})/dt^2$.

On the basis of this information, and of a dynamic model 712 of the robot arm, e.g. established using Newton-Euler equations, a vector \mathbf{C}_d is calculated representing the torques that each motor of the robot ought to be exerting in theory, assuming that the robot were not exerting any external force 713.

By taking the difference at 720, a vector \mathbf{C}_{ext} 800 is deduced that constitutes an estimate of motor torques corresponding to the forces exerted by the robot on its environment 800:

$$\mathbf{C}_{ext} = \mathbf{C}_m - \mathbf{C}_d$$

Figure 7 shows how the algorithm used by the path generator can be split up into four portions:

Starting with the tangential force setpoint 310, the external force 800, and the shape of the curve 320, a scalar error variable 420 is calculated at 410. The velocity and acceleration limits 330 and 340 are projected at 430 onto the curve 320. On the basis of the projected velocity and acceleration limits 470, and also on the basis of the error variable 420, a regulator 430 calculates the desired movement along the curve 440. The movement setpoints 500 of the robot are then calculated at 450 from the defined shape of the curve 320 and the defined movement along the curve 440.

Each of these four portions is described below:

A) *Calculating the error variable (410-420): movement in a straight line*

5 The calculation of the error variable 420 is described below for movement in a straight line. The calculations given below consist in calculating the projection onto the curve 320 of the difference between the force setpoint 310 and the external force 800.

10 During a movement in a straight line, the Cartesian position \underline{x} of the robot (a vector) can be written in parametric form as follows:

$$\underline{x} = \underline{x}_0 + s.(\underline{x}_1 - \underline{x}_0)$$

where \underline{x}_0 is the Cartesian position from which the movement starts, \underline{x}_1 is the Cartesian target position, and s is a scalar parameter lying in the range 0 to 1. The above equation can be rewritten in the following form:

(Eq. 1) $\underline{x} = f(s)$

By differentiating this equation, the following is obtained:

(Eq. 2) $\underline{v} = d\underline{x}/dt = df/ds.ds/dt$

In this equation, df/ds is a Cartesian vector tangential to the curve. The Cartesian vector:

$$\underline{u} = (df/ds) / \|df/ds\|$$

25 is unitary and tangential to the curve.

In this example, the force setpoint 300 is the norm of the Cartesian tangential force. The desired Cartesian force vector is thus given by:

(Eq. 3) $\underline{F}_{des} = f_{des}.\underline{u}$

30 The external force 800 is given by the vector of the joint torques \underline{C}_{ext} . This value can be transformed into Cartesian coordinates by using the following formula:

(Eq. 4) $\underline{F}_{ext} = \underline{J}^{-t}.\underline{C}_{ext}$

where \underline{J}^{-t} is the inverse of the transpose of the kinematic matrix of the robot.

The selected command variable 420, written ϵ , is the difference between the desired tangential force and

the external force, as projected onto the tangent of the path, i.e.:

$$\text{eps} = \mathbf{u}^t \cdot (\mathbf{F}_{\text{des}} - \mathbf{F}_{\text{ext}})$$

This calculation remains valid regardless of the
 5 function \underline{f} used (circle, spline, etc. ...), providing it can be differentiated, which means that its tangent can be defined at all points.

This calculation has been performed for the special case where the force setpoint 300 is the norm of a
 10 Cartesian force vector and the external force 800 is given in joint coordinates. By using appropriate change-of-coordinate formulae, e.g. replacing equation Eq. 3 and equation Eq. 4, it is possible to handle the situation in which this which is given in any coordinate system.

15

B) Calculating the error variable (410-420): for joint movement

When handling joint movement, the joint position \underline{q} of the robot can be written in parametric form as
 20 follows:

$$\underline{q} = \underline{q}_0 + s \cdot (\underline{q}_1 - \underline{q}_0)$$

where \underline{q}_0 is the joint position at the start of the movement, \underline{q}_1 is the target joint position, and \underline{s} is scalar parameter lying in the range 0 to 1. The above
 25 equation can be rewritten in the form:

$$\underline{q} = \underline{g}(s)$$

The corresponding Cartesian position is given by:

$$\underline{x} = \underline{k}(\underline{q}(s))$$

where \underline{k} is the direct geometrical model of the robot.

30 The joint movement can thus be handled like the Cartesian movement, by selecting the function \underline{f} such that:

$$\underline{f}(s) = \underline{k}(\underline{q}(s))$$

35 *C) Projecting the velocity and acceleration limits along the curve (460)*

In this example, the velocity limits are given in

Cartesian coordinates. In other words, it is desired that the norm of the velocity \underline{v} of the robot should be less than a given velocity v_{\max} , and the norm of the Cartesian acceleration $d\mathbf{v}/dt$ should be less than a given acceleration a_{\max} .

The following values are defined:

$$(\text{Eq. 5}) \quad (ds/dt)_{\max} = \min(v_{\max}/\|df/ds\|); \quad \sqrt{a_{\max}/\|df/ds\|}$$

$$(\text{Eq. 6}) \quad (d^2s/dt^2)_{\max} = (a_{\max} - \|df/ds\| \cdot (ds/dt)_{\max}) / \|d^2f/ds^2\|$$

in which c is a constant such that $0 < c < 1$

10 $\sqrt{}$ designates the square root function.

Assuming the following applies:

$$|ds/dt| \leq (ds/dt)_{\max}$$

$$|d^2s/dt^2| \leq (d^2s/dt^2)_{\max}$$

Then, from equation Eq. 2, the following applies:

$$\begin{aligned} 15 \quad \|\mathbf{v}\| &= \|df/ds \cdot ds/dt\| \\ &= \|df/ds\| \cdot \|ds/dt\| \\ &\leq \|df/ds\| \cdot (ds/dt)_{\max} \end{aligned}$$

Thus, from equation Eq. 5, the following applies:

$$\|\mathbf{v}\| \leq v_{\max}$$

20 In addition, by differentiating equation Eq. 2, the following is obtained:

$$\mathbf{a} = df/ds \cdot d^2s/dt^2 + d^2f/ds^2 \cdot (ds/dt)^2$$

whence

$$\begin{aligned} \|\mathbf{a}\| &= \|df/ds \cdot d^2s/dt^2 + d^2f/ds^2 \cdot (ds/dt)^2\| \\ 25 \quad &\leq \|df/ds\| \cdot \|d^2s/dt^2\| + \|d^2f/ds^2\| \cdot |ds/dt|^2 \\ &\leq \|df/ds\| \cdot (d^2s/dt^2)_{\max} + \|d^2f/ds^2\| \cdot (ds/dt)_{\max}^2 \end{aligned}$$

By using the definitions of equations Eq. 5 and Eq. 6, the following is obtained:

$$\|\mathbf{a}\| \leq a_{\max}$$

30 The values $(ds/dt)_{\max}$ and $(d^2s/dt^2)_{\max}$ thus define constraints on the derivatives of \underline{s} , which ensure that the Cartesian velocity and acceleration limits are satisfied. It can thus be considered that these are projections of the Cartesian velocity and acceleration limits onto the coordinates of \underline{s} .

Similarly, it is possible to have joint velocity and/or acceleration limits by using the function $h(s) =$

$k^{-1}(f(s))$ instead of the function \underline{f} in the preceding calculations.

D) Regulator (430)

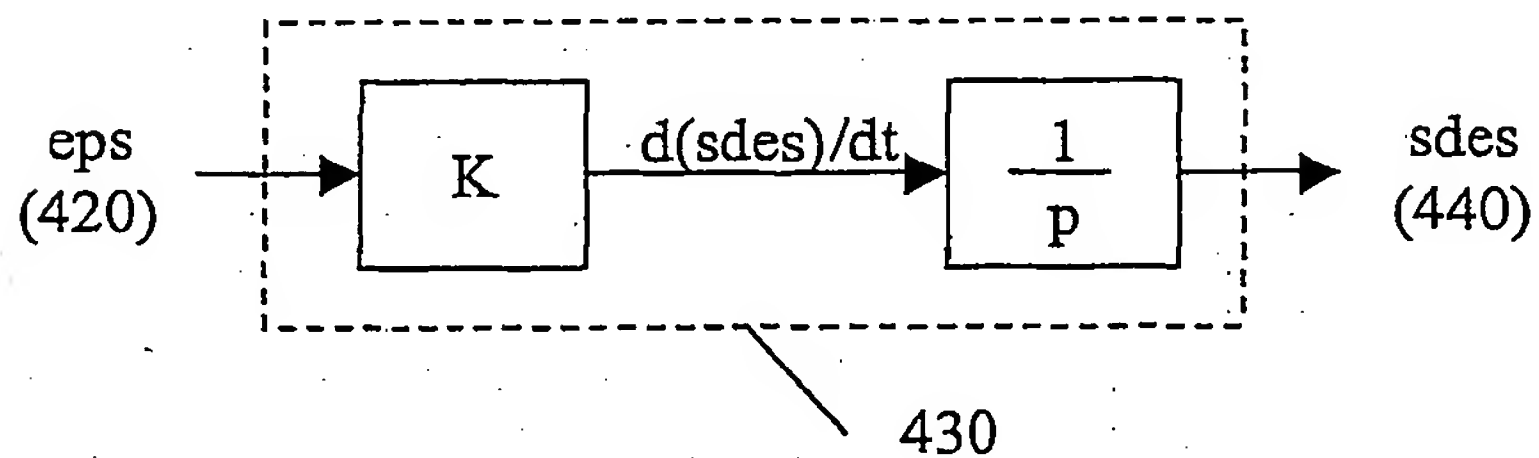
5 The function of the regulator 430 is to generate movements along the curve 440, so as to keep the error variable as close as possible to zero 420.

10 The regulator used is of the integral type. A "conventional" integral regulator can be expressed in the following form:

$$d(sdes)/dt = K.eps$$

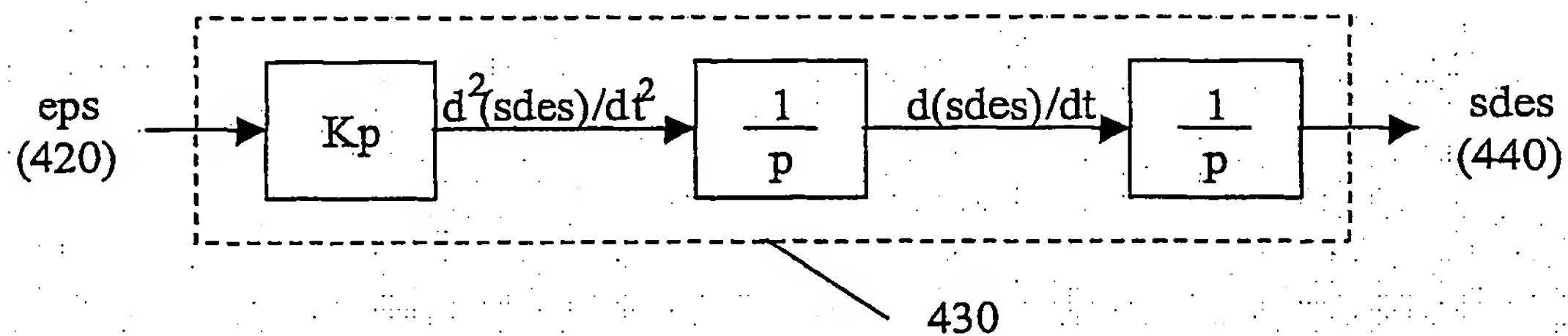
$$sdes = \int d(sdes)/dt . dt$$

15 where sdes is the desired position for the robot along the curve, expressed in the units of the parameter \underline{s} in equation Eq. 1, and K is a constant. This is shown in the following block diagram, where \underline{p} designates the Laplace variable:

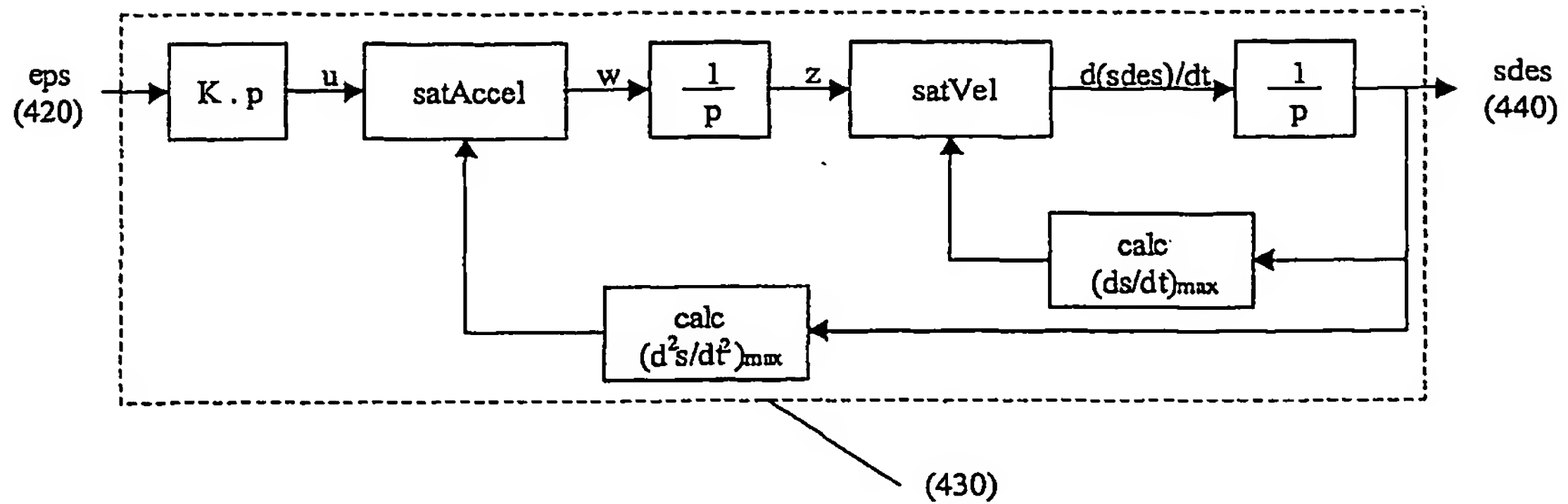


20

This diagram is equivalent to:



25 In order to include the velocity and acceleration limits, the diagram is modified as follows:



The block "calc $(ds/dt)_{\max}$ " and "calc $(d^2s/dt^2)_{\max}$ " calculates the speed of acceleration limits projected as a function of $sdes$ as given by equations Eq. 5 and Eq. 6 (df/ds and d^2f/ds^2 are evaluated for $s = sdes$). The block $satVel$ saturates the input velocity z as a function of $(ds/dt)_{\max}$. In other words, the following applies:

$$10 \quad d(sdes)/dt = \begin{cases} (ds/dt)_{\max} & \text{if } z > (ds/dt)_{\max} \\ z & \text{if } |z| \leq (ds/dt)_{\max} \\ -(ds/dt)_{\max} & \text{if } z < -(ds/dt)_{\max} \end{cases}$$

Similarly, $satAccel$ saturates the acceleration u as a function of $(d^2s/dt^2)_{\max}$, i.e.:

$$15 \quad w = \begin{cases} (d^2s/dt^2)_{\max} & \text{if } u > (d^2s/dt^2)_{\max} \\ u & \text{if } |u| \leq (d^2s/dt^2)_{\max} \\ -(d^2s/dt^2)_{\max} & \text{if } u < -(d^2s/dt^2)_{\max} \end{cases}$$

Under all circumstances, this scheme ensures that:

$$|d(sdes)/dt| < (ds/dt)_{\max}$$

which guarantees that the norm of the desired Cartesian velocity along the path is less than v_{\max} .

When the maximum velocity is not reached, then:

$$z = d(sdes)/dt$$

thus

$$w = d^2(sdes)/dt^2$$

25 and

$$|d^2(s_{des})/dt^2| < (d^2s/dt^2)_{\max}$$

which guarantees that the norm of the Cartesian acceleration is less than a_{\max} .

In a general context, the algorithm described
 5 ensures that the Cartesian velocity limit is always
 satisfied and that the acceleration limit is satisfied
 insofar as the maximum Cartesian velocity is not reached.
 For a straight line, when the velocity limit is reached,
 acceleration is zero, so the acceleration limit is
 10 satisfied.

E) Calculating (430) movement setpoints (500)

In this example, the movement setpoints 500 are the
 robot joint coordinates q_{des} . Starting from the desired
 15 position along the curve s_{des} , it is possible to
 calculate the corresponding Cartesian position setpoint
 by using the parametric equation for the curve as given
 by Eq. 1:

$$x_{des} = f(s_{des})$$

20 Thereafter, the joint coordinates are obtained using
 the robot inverse geometrical model written k^{-1} :

$$q_{des} = k^{-1}(x_{des})$$

If the movement setpoints were velocities, it would
 be possible to use the corresponding coordinate changes
 25 on velocities (equation Eq. 2 and the inverse of the
 robot kinematic matrix).